

09-05-00

EK2873847524

A

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Date - 8-31-00

Docket No. AUS9-2000-0295-US1

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of Inventor(s):

Viktors Berstis  
Lorin Evan UllmanFor: Method and Apparatus in a Data Processing System for Word Based Render  
Browser for Skimming or Speed Reading Web Pages

Enclosed are also:

- ☒ 21 Pages of Specification including an Abstract  
☒ 6 Pages of Claims  
☒ 12 Sheet(s) of Drawings  
☒ A Declaration and Power of Attorney  
☒ Form PTO 1595 and assignment of the invention to IBM Corporation

## CLAIMS AS FILED

FOR	Number Filed		Number Extra		Rate		Basic Fee (\$690)
Total Claims	36	-20 =	16	X	\$ 18	=	\$288
Independent Claims	7	-3 =	4	X	\$ 78	=	\$312
Multiple Dependent Claims	0			X	\$260	=	\$0
<b>Total Filing Fee</b>							<b>= \$1,290</b>

- ☒ Please charge \$1,290.00 to IBM Corporation, Deposit Account No. 09-0447.  
☒ The Commissioner is hereby authorized to charge payment of the following fees associated with the communication or credit any over payment to IBM Corporation, Deposit Account No. 09-0447. A duplicate copy of this sheet is enclosed.  
☒ Any additional filing fees required under 37CFR § 1.16.  
☒ Any patent application processing fees under 37CFR § 1.17.

Respectfully,

David A. Mims, Jr.

Reg. No. 32,708

Intellectual Property Law Dept.

IBM Corporation

11400 Burnet Road 4054

Austin, Texas 75758

Telephone: (512) 823-0950

JC887 U.S. PTO  
08/31/00JC853 U.S. PTO  
09/652365  
08/31/00

09-05-00

Docket No. AUS9-2000-0295-US1

**METHOD AND APPARATUS IN A DATA PROCESSING SYSTEM FOR WORD  
BASED RENDER BROWSER FOR SKIMMING OR  
SPEED READING WEB PAGES**

5 **BACKGROUND OF THE INVENTION**

**1. Technical Field:**

The present invention relates generally to an improved data processing system and in particular to a  
10 method and apparatus for handling documents in a data processing system. Still more particularly, the present invention provides a method and apparatus for modifying a web page to facilitate skimming or speed reading of the web page.

15 **2. Description of Related Art:**

The Internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer  
20 and the conversion of messages from the sending network to the protocols used by the receiving network (with packets if necessary). When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

25 The Internet has become a cultural fixture as a source of both information and entertainment. Many businesses are creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing  
30 other information seeking to engender brand loyalty. Many federal, state, and local government agencies are also

Docket No. AUS9-2000-0295-US1

employing Internet sites for informational purposes,  
particularly agencies which must interact with virtually  
all segments of society such as the Internal Revenue  
Service and secretaries of state. Providing informational  
5 guides and/or searchable databases of online public  
records may reduce operating costs. Further, the Internet  
is becoming increasingly popular as a medium for  
commercial transactions.

Currently, the most commonly employed method of  
10 transferring data over the Internet is to employ the World  
Wide Web environment, also called simply "the Web". Other  
Internet resources exist for transferring information,  
such as File Transfer Protocol (FTP) and Gopher, but have  
not achieved the popularity of the Web. In the Web  
15 environment, servers and clients effect data transaction  
using the Hypertext Transfer Protocol (HTTP), a known  
protocol for handling the transfer of various data files  
(e.g., text, still graphic images, audio, motion video,  
etc.). The information in various data files is formatted  
20 for presentation to a user by a standard page description  
language, the Hypertext Markup Language (HTML). In  
addition to basic presentation formatting, HTML allows  
developers to specify "links" to other Web resources  
identified by a Uniform Resource Locator (URL). A URL is  
25 a special syntax identifier defining a communications path  
to specific information. Each logical block of  
information accessible to a client, called a "page" or a  
"Web page", is identified by a URL. The URL provides a  
universal, consistent method for finding and accessing  
30 this information, not necessarily for the user, but mostly  
for the user's Web "browser". A browser is a program  
capable of submitting a request for information identified

Docket No. AUS9-2000-0295-US1

by an identifier, such as, for example, a URL. A user may enter a domain name through a graphical user interface (GUI) for the browser to access a source of content. The domain name is automatically converted to the Internet Protocol (IP) address by a domain name system (DNS), which is a service that translates the symbolic name entered by the user into an IP address by looking up the domain name in a database.

More and more businesses and individuals are placing web pages and other types of content on the Internet. The amount of information available on the Internet has become extremely large. Search engines are present, which identify content that a user may desire. These search engines, however, may return hundreds of pages in response to a query. This amount of information often times causes user frustration because of the large number of web pages that a user must review to find desired content. Some search engines may organize the content based on how closely the content corresponds to the query or based on subject matter. Still, the user must review the pages until the desired content is found.

Therefore, it would be advantageous to have an improved method and apparatus for quickly reviewing the content of web pages.

Docket No. AUS9-2000-0295-US1

### SUMMARY OF THE INVENTION

The present invention provides a method and  
5 apparatus in a data processing system for modifying  
content of a document. A request is received for  
modified content. The document is compressed using a set  
of rules, wherein selected content in the document is  
removed to increase a speed at which a user can read the  
10 document.

Docket No. AUS9-2000-0295-US1

### BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

**Figure 1** depicts a pictorial representation of a distributed data processing system in which the present invention may be implemented;

**Figure 2** is a block diagram of a data processing system that may be implemented as a server, such as a server in **Figure 1**, in accordance with a preferred embodiment of the present invention;

**Figure 3** is a block diagram illustrating a data processing system in which the present invention may be implemented;

**Figure 4** is a diagram illustrating data flow in a word based speed reading system in accordance with a preferred embodiment of the present invention;

**Figure 5** is a diagram of a graphical user interface (GUI) for generating rules for word based speed reading in accordance with a preferred embodiment of the present invention;

**Figure 6** is an example GUI for keeping words in accordance with a preferred embodiment of the present invention;

**Figure 7** is a flowchart of a process used by a word policy manager to generate word policies and policy

Docket No. AUS9-2000-0295-US1

helpers in accordance with a preferred embodiment of the present invention;

5       **Figure 8** is a flowchart of a process used to generate a modified document in accordance with a preferred embodiment of the present invention;

**Figure 9** is an illustration of a word policy in accordance with a preferred embodiment of the present invention;

10       **Figure 10** is a diagram illustrating a compression helper method in accordance with a preferred embodiment of the present invention; and

**Figure 11** is a diagram illustrating a code segment for modifying a document in accordance with a preferred embodiment of the present invention.

Docket No. AUS9-2000-0295-US1

### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, Figure 1 depicts a  
5 pictorial representation of a distributed data processing  
system in which the present invention may be implemented.  
Distributed data processing system 100 is a network of  
computers in which the present invention may be  
implemented. Distributed data processing system 100  
10 contains a network 102, which is the medium used to  
provide communications links between various devices and  
computers connected together within distributed data  
processing system 100. Network 102 may include permanent  
connections, such as wire or fiber optic cables, or  
15 temporary connections made through telephone connections.

In the depicted example, a server 104 is connected to  
network 102 along with storage unit 106. In addition,  
clients 108, 110, and 112 also are connected to network  
102. These clients 108, 110, and 112 may be, for example,  
20 personal computers or network computers. For purposes of  
this application, a network computer is any computer,  
coupled to a network, which receives a program or other  
application from another computer coupled to the network.  
In the depicted example, server 104 provides data, such as  
25 boot files, operating system images, and applications to  
clients 108-112. Clients 108, 110, and 112 are clients to  
server 104. Distributed data processing system 100 may  
include additional servers, clients, and other devices not  
shown. In the depicted example, distributed data  
30 processing system 100 is the Internet with network 102  
representing a worldwide collection of networks and  
gateways that use the TCP/IP suite of protocols to

Docket No. AUS9-2000-0295-US1

communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, distributed data processing system 100 also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). Figure 1 is intended as an example, and not as an architectural limitation for the present invention.

Referring to Figure 2, a block diagram of a data processing system that may be implemented as a server, such as server 104 in Figure 1, is depicted in accordance with a preferred embodiment of the present invention. Data processing system 200 may be a symmetric multiprocessor (SMP) system including a plurality of processors 202 and 204 connected to system bus 206. Alternatively, a single processor system may be employed. Also connected to system bus 206 is memory controller/cache 208, which provides an interface to local memory 209. I/O bus bridge 210 is connected to system bus 206 and provides an interface to I/O bus 212. Memory controller/cache 208 and I/O bus bridge 210 may be integrated as depicted.

Peripheral component interconnect (PCI) bus bridge 214 connected to I/O bus 212 provides an interface to PCI local bus 216. A number of modems may be connected to PCI bus 216. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers 108-112 in

Docket No. AUS9-2000-0295-US1

Figure 1 may be provided through modem 218 and network adapter 220 connected to PCI local bus 216 through add-in boards.

Additional PCI bus bridges 222 and 224 provide  
5 interfaces for additional PCI buses 226 and 228, from which additional modems or network adapters may be supported. In this manner, data processing system 200 allows connections to multiple network computers. A memory-mapped graphics adapter 230 and hard disk 232 may  
10 also be connected to I/O bus 212 as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate that the hardware depicted in Figure 2 may vary. For example, other peripheral devices, such as optical disk  
15 drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

The data processing system depicted in Figure 2 may  
20 be, for example, an IBM RISC/System 6000 system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system.

With reference now to Figure 3, a block diagram  
25 illustrating a data processing system in which the present invention may be implemented. Data processing system 300 is an example of a client computer. Data processing system 300 employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted  
30 example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard

Docket No. AUS9-2000-0295-US1

Architecture (ISA) may be used. Processor 302 and main memory 304 are connected to PCI local bus 306 through PCI bridge 308. PCI bridge 308 also may include an integrated memory controller and cache memory for processor 302.

5 Additional connections to PCI local bus 306 may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter 310, SCSI host bus adapter 312, and expansion bus interface 314 are connected to PCI local bus 306 by direct  
10 component connection. In contrast, audio adapter 316, graphics adapter 318, and audio/video adapter 319 are connected to PCI local bus 306 by add-in boards inserted into expansion slots. Expansion bus interface 314 provides a connection for a keyboard and mouse adapter  
15 320, modem 322, and additional memory 324. Small computer system interface (SCSI) host bus adapter 312 provides a connection for hard disk drive 326, tape drive 328, and CD-ROM drive 330. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in  
20 connectors.

An operating system runs on processor 302 and is used to coordinate and provide control of various components within data processing system 300 in Figure 3. The operating system may be a commercially available operating  
25 system, such as Windows 2000, which is available from Microsoft Corporation. An object oriented programming system such as Java may run in conjunction with the operating system and provides calls to the operating system from Java programs or applications executing on  
30 data processing system 300. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system,

Docket No. AUS9-2000-0295-US1

the object-oriented operating system, and applications or programs are located on storage devices, such as hard disk drive 326, and may be loaded into main memory 304 for execution by processor 302.

5        Those of ordinary skill in the art will appreciate that the hardware in **Figure 3** may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash ROM (or equivalent nonvolatile memory) or optical disk drives and the like, may be used  
10 in addition to or in place of the hardware depicted in **Figure 3**. Also, the processes of the present invention may be applied to a multiprocessor data processing system.

For example, data processing system 300, if  
15 optionally configured as a network computer, may not include SCSI host bus adapter 312, hard disk drive 326, tape drive 328, and CD-ROM 330, as noted by dotted line 332 in **Figure 3** denoting optional inclusion. In that case, the computer, to be properly called a client  
20 computer, must include some type of network communication interface, such as LAN adapter 310, modem 322, or the like. As another example, data processing system 300 may be a stand-alone system configured to be bootable without relying on some type of network communication interface,  
25 whether or not data processing system 300 comprises some type of network communication interface. As a further example, data processing system 300 may be a Personal Digital Assistant (PDA) device, which is configured with ROM and/or flash ROM in order to provide non-volatile  
30 memory for storing operating system files and/or user-generated data.

Docket No. AUS9-2000-0295-US1

The depicted example in **Figure 3** and above-described examples are not meant to imply architectural limitations. For example, data processing system **300** also may be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing system **300** also may be a kiosk or a Web appliance.

The present invention provides an improved method, apparatus, and computer implemented instructions for allowing a user to skim or speed read a web page. The present invention recognizes that no mechanism is present to compress content in a document, such as a web page, without degrading readability of the message by the user and without changing the physical and special characteristics of the original document. Currently available compression techniques are mainly used to improved storage of data, communication time, and performance. These mechanisms, however, change the physical relationships between text and graphical objects on a web page.

The mechanism of the present invention modifies content to display some words while making other words invisible. A content's message to the user is maintained to retain the meaning of the content, mainly the amount of information displayed. Just as a public speaker modifies his/her delivery based on his/her audience, this mechanism compresses content based on is user's interests or abilities obtained via configuration content before the content is displayed.

The mechanism of the present invention allows for skimming or speed reading of a web page by making trivial words invisible to increase the speed by which information on a web page may be read or comprehended by

Docket No. AUS9-2000-0295-US1

user. The amount of improvement in speed at which  
information can be read by user depends on the cognitive  
skills of the particular user. For example, words that  
are made invisible or stay visible may be selected based  
5 on the educational level of difficulty, the length, or  
the function of the word in the sentence. Other  
information, such as the number of pages reviewed per  
minute by user, word compression statistics, and the  
amount of data skimmed may be used in identifying words  
10 that are to be visible or made invisible.

Turning next to **Figure 4**, a diagram illustrating  
data flow in a word based speed reading system is  
depicted in accordance with a preferred embodiment of the  
present invention. In this example, server **400** is an  
15 HTTP web server and may be implemented using a server,  
such as data processing system **200** in **Figure 2**. In this  
example, server **400** may modify or alter content for use  
in a web browser **402**, which is located on a client, such  
as client **110** in **Figure 1**. Browser **402** make be  
20 implemented using know browser applications, such  
Internet Explorer, which is available from Microsoft  
Corporation.

The various settings or rules used to alter content  
may be set using browser **404**. This browser as browser  
25 **402** or a different browser. Also, browser **404** may be  
located on the same client, such as client **110**, or on a  
different client, such as client **108** in **Figure 1**.

In particular, word based speed reading graphical  
user interface (GUI) **406** is a user interface used to set  
30 up and alter rules in server **400** used to modify or alter  
content in a document, such as a web page. In these

Docket No. AUS9-2000-0295-US1

examples, the rules include rules to delete words 408, rules to keep words 410, and rules to replace words 412. Rules to delete words 408 are used to identify words that can be deleted or made invisible in a document. Rules to  
5 keep words 410 are used to identify words in a document that should be retained or made visible. Rules to replace words 412 are used to identify words that may be replaced with alternate words to make the readability or comprehension of the word easier. Word policy manager  
10 414 may take selections of these rules made through word based speed reading GUI 406 to generate a profile or word policy for a particular user. In the depicted examples, the word policy is a data structure containing the various rules and words received and selected from word  
15 based speed reading GUI 406. Different word policies and users associated with these word policies are stored in word based speed reading database 416.

In this example, browser 402 sends a request to modify a web page for a user. This request is received  
20 by word compression engine 418 in server 400. In this example, the request includes a user identification as well as the web page to be modified. The web page is sent from a content render engine 420 in web browser 402 to word compression engine 418. Based on the  
25 identification received in the request, a word policy for a user is retrieved from word based speed reading database 416 by word policy manager 414 and for use by a policy helper method, which aids word compression engine 418 in analyzing various words in the web page to modify  
30 or compress the web page to facilitate faster comprehension of the content in the web page. In these

Docket No. AUS9-2000-0295-US1

examples, the policy helper method used by word compression engine 418 is a compression helper method. Using this policy helper method, word compression engine 418 modifies the web page to generate modified or compressed content. This modified content is then returned to content render engine 420 for display to the user. In the depicted examples, the modified or compressed content retains the physical and special characteristics of the original document.

Turning next to **Figure 5**, a diagram of graphical user interface GUI for generating rules for word based speed reading is depicted in accordance with a preferred embodiment of the present invention. GUI 500 is an example of a GUI that may be presented to a user to generate and select rules for a word policy. In this example, GUI 500 includes buttons, 502, 504, and 506 for a keep key words GUI, a remove words GUI and replace words GUI, respectively. Selection of one of these buttons result in an appropriate GUI to be displayed for generating or modifying a word policy.

Turning next to **Figure 6**, an example GUI for keeping words is depicted in accordance with a preferred embodiment of the present invention. Keep words GUI 600 may be presented in a variety of ways to a user. In this example, an HTML page is used to enter selections and data for words that are to be kept in a document. Fields 602 and 604 provide for input regarding the number of letters per word and limitations to the content. As illustrated, the default for an number of words is three in field 602 and content is not limited to one page as shown in field 604. Fields 606-612 allow for words to be

Docket No. AUS9-2000-0295-US1

retained based on a word attribute. In these examples, the attributes are bold, italics, underlined, and link. The default is to keep words with these attributes. Keep words GUI 600 also allows for words to be kept or

5 retained in the document based on the number of syllables in a word using field 614. In this example, the number of syllables is three as a default value.

Next, field 616 allows for words to be retained by grade level. The default here is for the tenth grade.

10 When checking for words by grade level, much like known systems used in checking spelling and checking grammar editors. Additionally, field 618 allows a list of words to be entered for a user in which these words will be retained if present in the document. Field 620 allows  
15 for a list of links to be entered in which these links will be retained. The list of links may be specified in a number of different ways. For example, links starting with a particular word or containing a particular word may be ones that are retained.

20 Turning next to **Figure 7**, a flowchart of a process used by a word policy manager to generate word policies and policy helpers is depicted in accordance with a preferred embodiment of the present invention. In this example, the illustrated process may be implemented in a  
25 word policy manager, such as word policy manager 414 in **Figure 4**.

The process begins by obtaining configuration data (step 700). This configuration data may be received from a web browser through an interface, such as word based  
30 speed reading GUI 406. This GUI displays appropriate pages for entry or selection of rules used to keep words,

Docket No. AUS9-2000-0295-US1

remove word, and/or replace words. Based on the configuration data obtained, word data is stored in a remove rules object, such as rules to delete words 408 in Figure 4 (step 702). Next, data obtained from the interface is stored in a replace rules object, such as rules to replace words 412 in Figure 4 (step 704). Configuration data also may be stored in a keep rules object, such as rules to keep words 410 in Figure 4 (step 706). The data stored in these objects is retrieved and a compression helper method is then created (step 708). The compression helper is then stored in a database along with a user key (step 710) with the process terminating thereafter. In this example, the database may be word based speed reading database 416 in Figure 4.

Turning next to Figure 8, a flowchart of a process used to generate a modified document is depicted in accordance with a preferred embodiment of the present invention. The process in Figure 8 may be implemented in word compression engine 418 in Figure 4. The process begins by receiving a request for modified content from a web browser (step 800). In this example, the request includes an identification of a user as well as the unmodified content, such as one or more web pages. Next, a compression helper method is received (step 802). This method in these examples is received from a word policy manager, such as word policy manager 414 in Figure 4. The content is then compressed using the compression helper method (step 804). In compressing the content, in these examples, the compression consists of displaying some words while other words are deleted or undisplayed. In deleting words, the physical and spatial

Docket No. AUS9-2000-0295-US1

characteristics of the original document remain intact in these examples. The modified content is then returned to the web browser (step 806) with the process terminating thereafter.

5           With reference now to **Figure 9**, an illustration of a word policy is depicted in accordance with a preferred embodiment of the present invention. Data structure **900** is an example of a word policy, representing rules and words that are to be considered in modifying or  
10       compressing a document. Data structure **900** in this example is implemented as a class that provides the different rules as well as a list of words to be deleted. Data structure **900** also references a word list, which is  
15       used by the policy helper method to aid word compression engine **418** in analyzing words to modify the web page. The word list represents words that should be removed or made invisible in the document in this example. Of course, data structure **900** may contain integers or other symbols representing words to be deleted.

20           Similar data structures are used to represent words to be kept or replaced. When replacing words, the data structure may also include the words or symbols for words to be replaced as well as the replacement words.

Turning next to **Figure 10**, a diagram illustrating a  
25       compression helper method is depicted in accordance with a preferred embodiment of the present invention. In this example, compression helper method **1000** takes the form of a class, which aids a compression engine, such as word compression engine **418** to analyze words. In this  
30       example, the method is implemented as a Java method.

With reference now to **Figure 11**, a diagram

Docket No. AUS9-2000-0295-US1

illustrating a code segment for modifying a document is depicted in accordance with a preferred embodiment of the present invention. Method 1100 illustrates example code used to receive content for modification, modifying the  
5 content, and placing the modified content in a file. In these examples, method 1100 is a Java method. The code in method 1100 creates a compressionHelpers which use query the DeleteWord, ReplaceWord and KeepWord classes in order to determine how to whether or not to display the  
10 word (or a replacement).

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of  
15 the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the  
20 distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications  
25 links using transmission forms, such as, for example, radio frequency and light wave transmissions. The computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

30 The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the

Docket No. AUS9-2000-0295-US1

invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, although the processes for modifying a document are illustrated as being located in  
5 a web server, these processes may be implemented in a number of different places. For example, these processes may be located on the same data processing system as the web browser. Further, these processes also may be integrated as part of a web browser. The processes may  
10 be implemented as a plug-in or filter for the web browser in such an implementation. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the  
15 invention for various embodiments with various modifications as are suited to the particular use contemplated.

Docket No. AUS9-2000-0295-US1

**CLAIMS:**

What is claimed is:

- 5 1. A method in a data processing system for modifying content of for a document, the method comprising:  
receiving request for modified content; and  
compressing the document using a set of rules,  
wherein selected content in the document is removed to  
10 increase a speed at which a user can read the document.
2. The method of claim 1, wherein the document is a web page.
- 15 3. The method of claim 1, wherein the document is a hypertext markup language document.
4. The method of claim 1, wherein the receiving step and the compressing step are performed in a server data  
20 processing system.
5. The method of claim 1, wherein the receiving step and the compressing step are performed in a client data processing system.
- 25 6. The method of claim 1, wherein the set of rules includes rules to delete words.
7. The method of claim 1, wherein the set of rules  
30 includes rules to include words.
8. The method of claim 1, wherein the set of rules includes rules to replace words.

Docket No. AUS9-2000-0295-US1

9. A method in a data processing system for altering content for a web page containing a set of words, the method comprising:

5       receiving a request to alter the web page; and  
      reducing the set of words in the web page to generate a modified web page, wherein the set of words is reduced using a set of rules and wherein the set of word in the modified web page retains key words allowing  
10   identification of the content of the web page.

10. The method of claim 9, wherein the web page is a hypertext markup language document.

15 11. The method of claim 9, wherein the receiving step and the reducing step are performed in a server data processing system.

20 12. The method of claim 9, wherein the receiving step and the reducing step are performed in a client data processing system.

25 13. The method of claim 9, wherein the set of rules includes rules to delete words.

14. The method of claim 9, wherein the set of rules includes rules to include words.

30 15. The method of claim 9, wherein the set of rules includes rules to replace words.

16. A data processing system comprising:

Docket No. AUS9-2000-0295-US1

a bus system;

a communications adapter connected to the bus,  
wherein the communications adapter provides for data  
transfer to and from the data processing system;

5 a memory connected to the bus system, wherein the  
memory includes a set of instructions; and

a processor unit connected to the bus, wherein the  
processor unit executes the set of instructions to a  
receiving a request to alter a web page and reduce the  
10 set of words in the web page to generate a modified web  
page, wherein the set of words is reduced using a set of  
rules and wherein the set of word in the modified web  
page retains key words allowing identification of the  
content of the web page.

15

17. The data processing system of claim 16, wherein the  
bus system includes a primary bus and a secondary bus.

18. The data processing system of claim 16, wherein the  
20 processing unit comprises one processor.

19. The data processing system of claim 16, wherein the  
processing unit comprises a plurality of processors.

25 20. A data processing system for modifying content of  
for a document, the data processing system comprising:

receiving means for receiving request for modified  
content; and

compressing means for compressing the document using  
30 a set of rules, wherein selected content in the document  
is removed to increase a speed at which a user can read  
the document.

Docket No. AUS9-2000-0295-US1

21. The data processing system of claim 20, wherein the document is a web page.

5 22. The data processing system of claim 20, wherein the document is a hypertext markup language document.

23. The data processing system of claim 20, wherein the receiving means and the compressing means are located in  
10 a server data processing system.

24. The data processing system of claim 20, wherein the receiving means and the compressing means are located in  
15 a client data processing system.

25. The data processing system of claim 20, wherein the set of rules includes rules to delete words.

26. The data processing system of claim 20, wherein the  
20 set of rules includes rules to include words.

27. The data processing system of claim 20, wherein the set of rules includes rules to replace words.

25 / 28. A data processing system for altering content for a web page containing a set of words, the data processing system comprising:

receiving means for receiving a request to alter the web page; and

30 reducing means for reducing the set of words in the web page to generate a modified web page, wherein the set of words is reduced using a set of rules and wherein the

Docket No. AUS9-2000-0295-US1

set of word in the modified web page retains key words  
allowing identification of the content of the web page.

29. The data processing system of claim 28, wherein the  
5 web page is a hypertext markup language document.

30. The data processing system of claim 28, wherein the  
receiving means and the reducing means are located in a  
server data processing system.

10

31. The data processing system of claim 28, wherein the  
receiving means and the reducing means are located in a  
client data processing system.

15 32. The data processing system of claim 28, wherein the  
set of rules includes rules to delete words.

33. The data processing system of claim 28, wherein the  
set of rules includes rules to include words.

20

34. The data processing system of claim 28, wherein the  
set of rules includes rules to replace words.

25 ~~35~~. A computer program product in a computer readable  
medium for use in a data processing system for modifying  
content of for a document, the computer program product  
comprising:

first instructions for receiving request for  
modified content; and

30 second instructions for compressing the document  
using a set of rules, wherein selected content in the  
document is removed to increase a speed at which a user

Docket No. AUS9-2000-0295-US1

can read the document.

36. A computer program product in a computer readable medium for use in a data processing system for altering content for a web page containing a set of words, the computer program product comprising:
- first instructions for receiving a request to alter the web page; and
  - second instructions for reducing the set of words in the web page to generate a modified web page, wherein the set of words is reduced using a set of rules and wherein the set of word in the modified web page retains key words allowing identification of the content of the web page.

Docket No. AUS9-2000-0295-US1

**ABSTRACT OF THE DISCLOSURE**

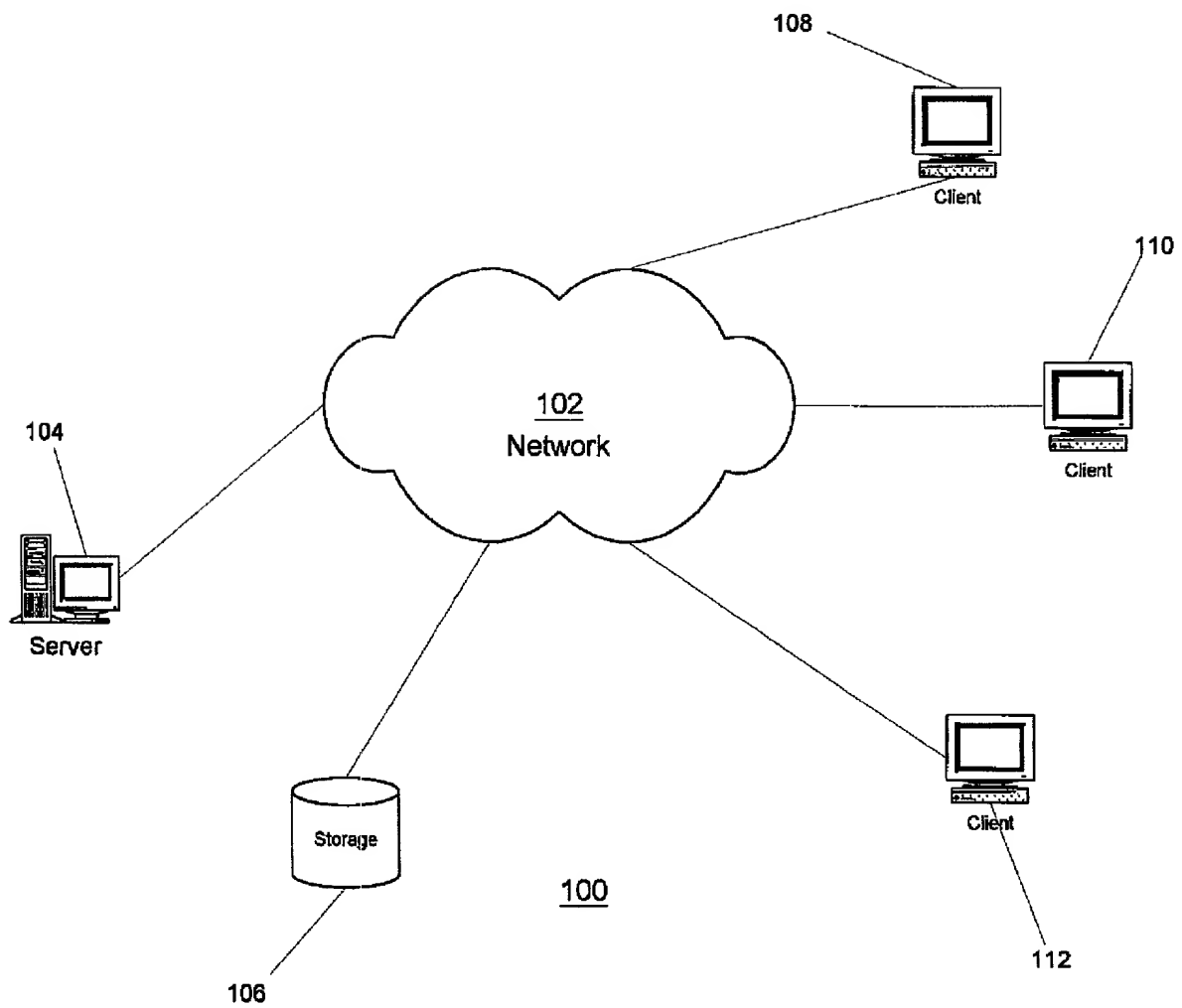
**METHOD AND APPARATUS IN A DATA PROCESSING SYSTEM FOR WORD  
5                   BASED RENDER BROWSER FOR SKIMMING OR  
                  SPEED READING WEB PAGES**

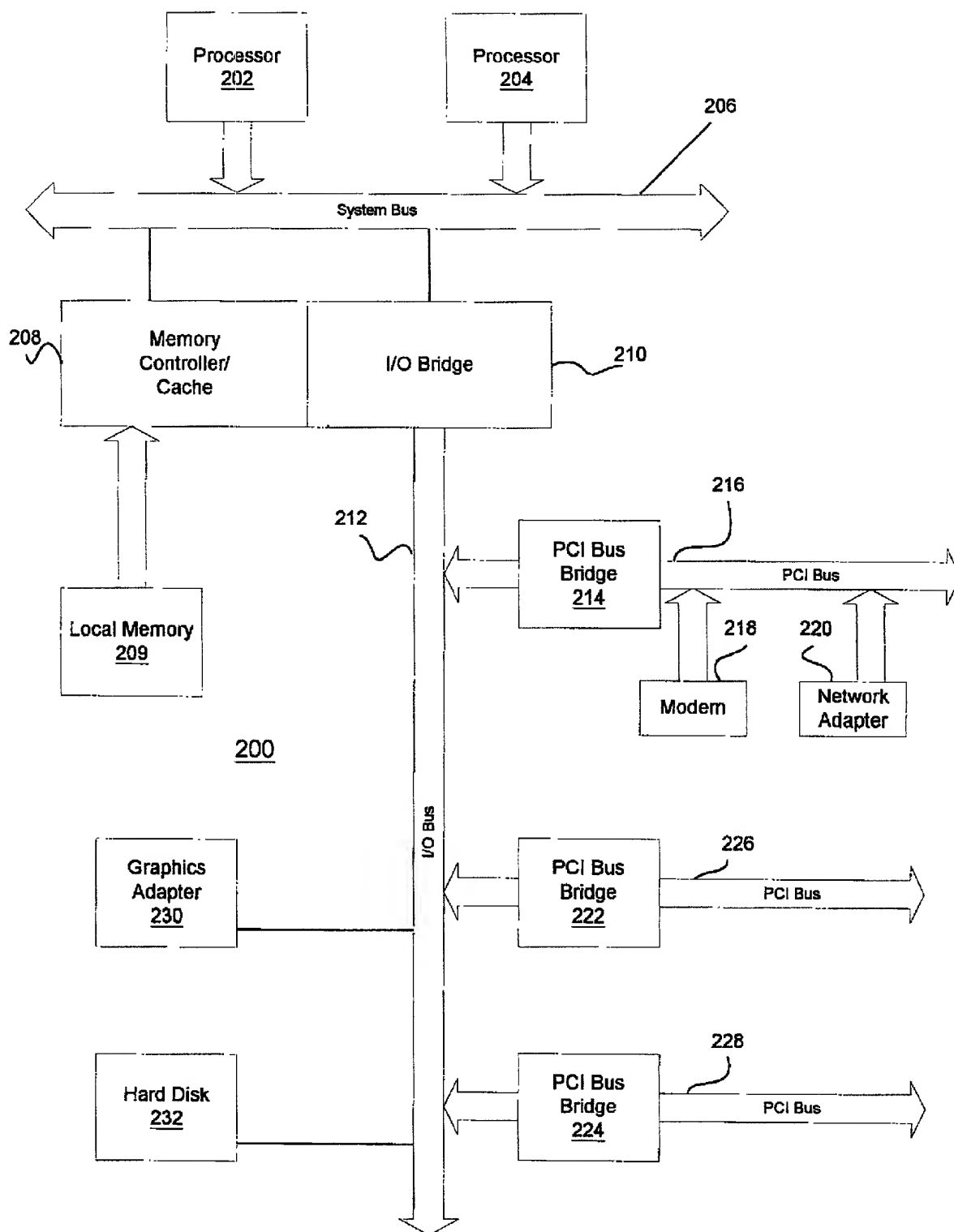
10           A method and apparatus in a data processing system  
for modifying content of a document. A request is  
received for modified content. The document is  
compressed using a set of rules, wherein selected content  
in the document is removed to increase a speed at which a  
user can read the document.

15

# Figure 1

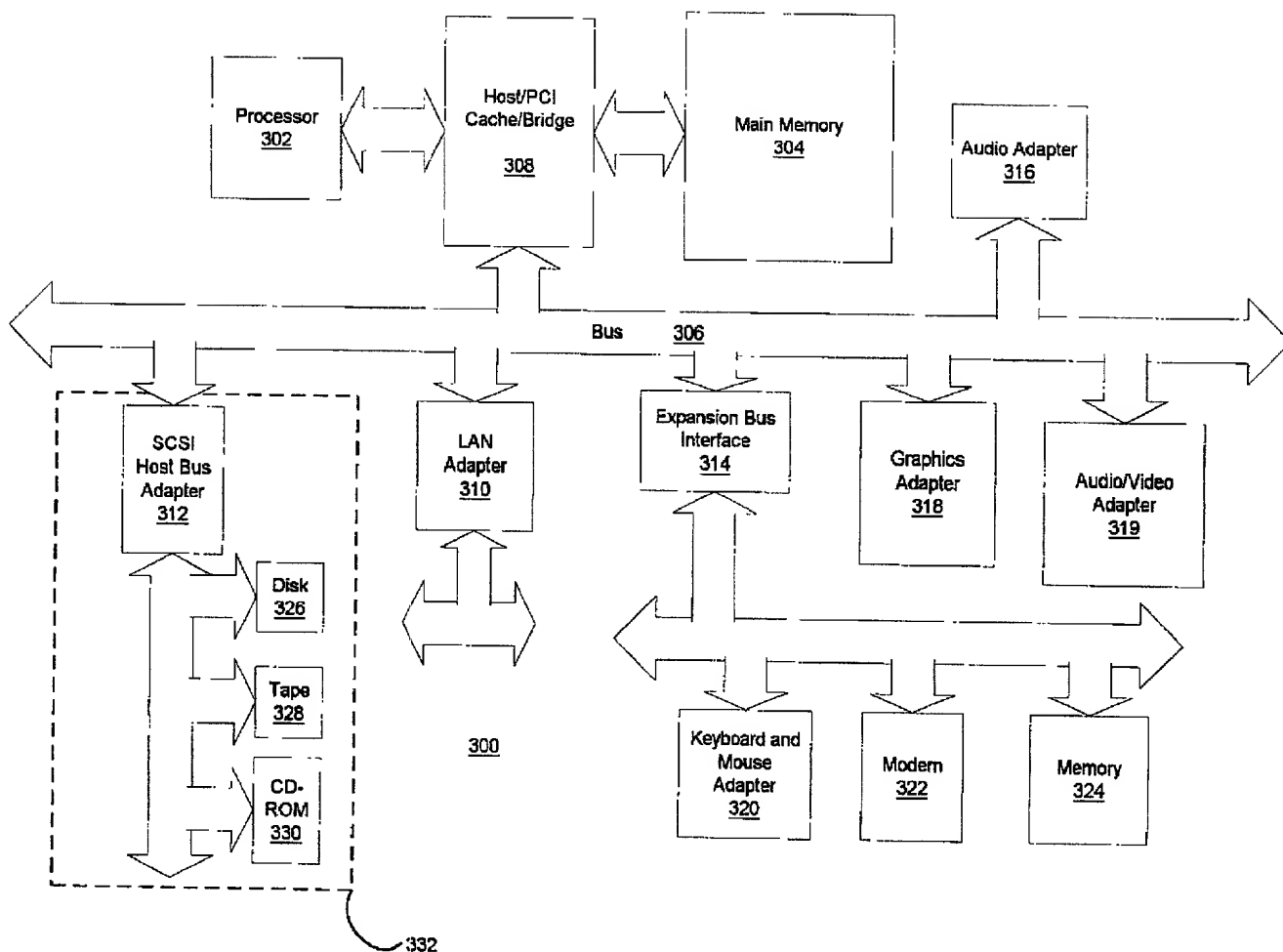
AUS9-2000-0295-US1





**Figure 2**

AUS9-2000-0295-US1

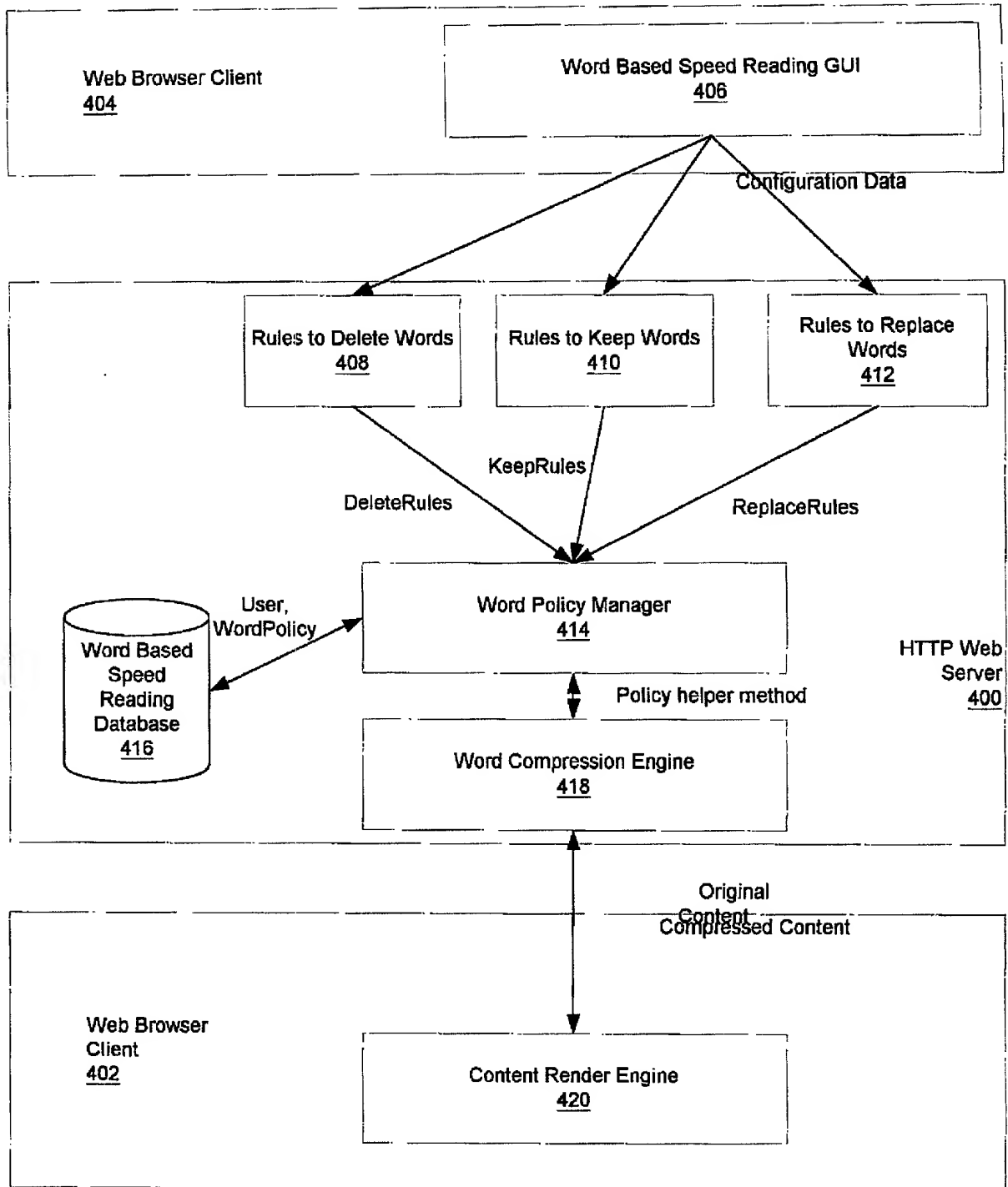


**Figure 3**

AUS9-2000-0295-US1

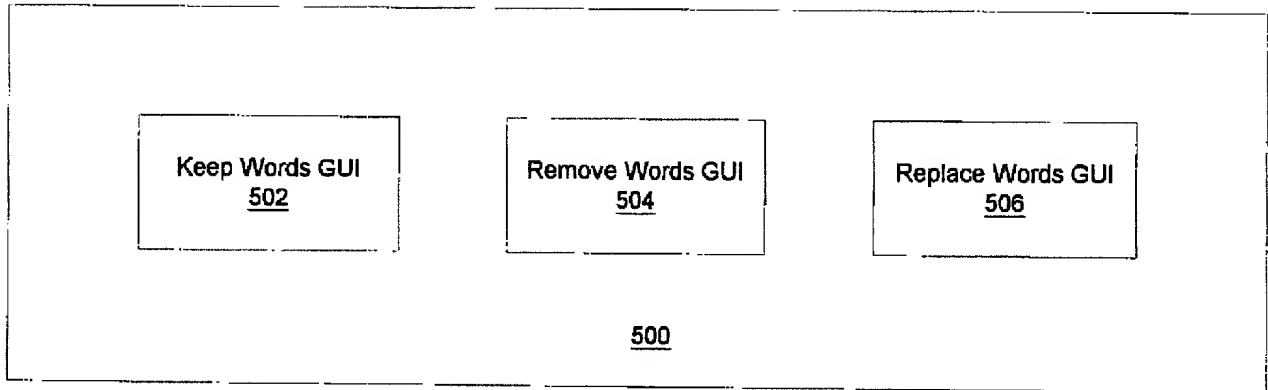
# Figure 4

AUS9-2000-0295-US1



# Figure 5

AUS9-2000-0295-US1



# Figure 6

AUS9-2000-0295-US1

Keep By Length      Number of Letters per Word  
Limit Content to One Page

3	602
no	604

Keep By Word  
Attribute

Bold  
Italic  
Underline  
Link

yes	606
yes	608
yes	610
yes	612

Keep By Number of Syllables

no, 3	614
-------	-----

Keep By Grade Level

10th	616
------	-----

Keep List of Words

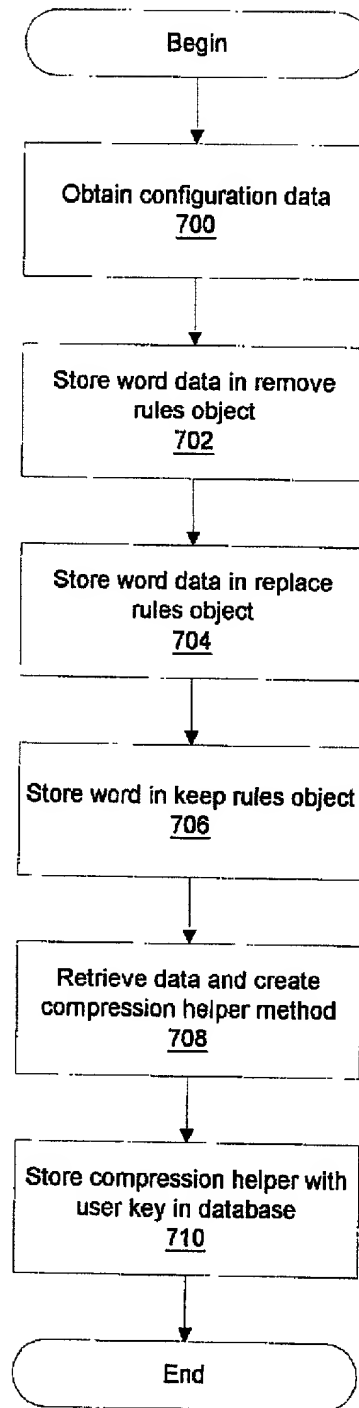
not, tennis, stock, new york	618
---------------------------------	-----

Keep List of Links

Starts With... Contains the word...	620
--	-----

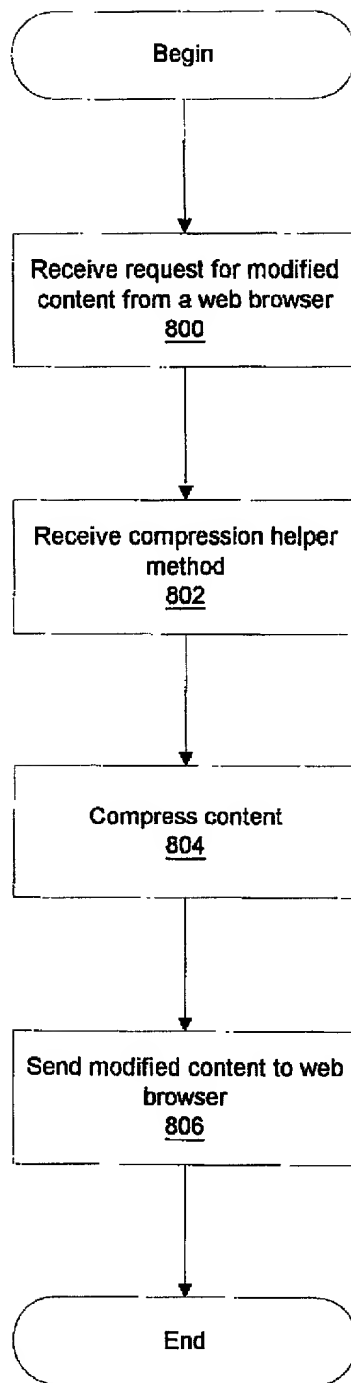
# Figure 7

AUS9-2000-0295-US1



# Figure 8

AUS9-2000-0295-US1



# Figure 9A

AUS9-2000-0295-US1

900

```
DeleteWord { // used if delete words defined in GUI

//init constructor
    DeleteWord (int length, int syllable, int
Difficulty, int attribute)

    //data
    int byLength;
    int bySyllable;
    int byDifficulty;
    int byAttribute;
    Vector removeFullWords;
    Vector removeStartsWithWords;

    //methods
    void setFullWord (String word) {}
    void setStartsWith (String word) {}
    boolean deleteWord (String word) {
        //compare length, difficulty, attributes
        //compare with removeFullWords list
        //compare with removeStartsWithWords list
        //return true or false

    void setWord (String word) {
        //used by the GUI to add words to delete (or extended
by Keep Class below)
        //add to Vector
    }
    void setStartsWithWord (String word) {
        //used by the GUI to add words
        //add to Vector
    }
}
```

---

# Figure 9B

AUS9-2000-0295-US1

```
ReplaceWords { // used if replace words defined in GUI
    //data
    Hashtable wordsToBeReplaced
        //key=word to be replaced
        //value=replacement word
    //methods
    boolean replace (String word) {
        //check hash to decide return true
    }
    void setReplacement (String wordToBeReplace,
replacement) {
        //used by the GUI to add words
    }

    String getReplacement (String word) { }
}

KeepWords { // used if keep words defined in GUI
    extends DeleteWords

    //init constructor
    KeepWords (int length, int syllable, int
Difficulty, int attribute)

    //methods

    boolean keepWord (String word ) {
        //compare length, difficulty, attributes
        //compare with removeFullWords list
        //compare with removeStartsWithWords list
        //return true or false
    }
}
```

# Figure 10

AUS9-2000-0295-US1

1000

```
CompressionHelper {

//methods
String getReplacement (word){ };

    int getNumSyllables (String word) { return numberOfSyllables }
    int getWordLength (String word) { return wordLength }
    int getDifficulty (String word) { return GradeLevelDifficulty }
    int getAttributes (String word) //bold=1, underline=2, italic=3, etc.
    { return VectorOfAttributes

//data
boolean isPartOfWordRemoveList = DeleteWords.deleteWord (word);
boolean isPartOfWordKeepList = KeepWords.keepWord (word);
boolean isPartOfWordReplacelist = ReplaceWords.deleteWord (word);

//init Constructors
CompressionHelper(String word) { };
CompressionHelper(String [ ] words) { };

}
```

## Figure 11

AUS9-2000-0295-US1

1100

```
//String getModifiedContent (String OriginalContentInFileFormat) {  
    //create InputStream from OriginalContentFile  
    //create InputStream for ModifiedContentOutputFile  
    //loop through all words  
        ifWordOnDeleteList //create CompressionHelper classes  
with each word to be analyzed in parallel with reading the unmodified  
file content. After caching the compression helpers away, the boolean  
falgs can be used to determine how the modified content is rendered  
(word removed, word replaced, word remains intact).  
        and NOT on isWordOnKeepList OR isWordOnReplaceList  
            //delete word  
            //break next word  
        else IfOnKeepList AND NOT on WordReplaceList  
            //break next word  
        else //OnWordReplacelist  
            //replace word  
  
    Write result to ModifiedContentOutoutFile  
}
```

DECLARATION AND POWER OF ATTORNEY FOR  
PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND APPARATUS IN A DATA PROCESSING SYSTEM FOR WORD BASED RENDER BROWSER  
FOR SKIMMING OR SPEED READING WEB PAGES

the specification of which (check one)

X is attached hereto.

— was filed on \_\_\_\_\_  
as Application Serial No. \_\_\_\_\_  
and was amended on \_\_\_\_\_  
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Priority Claimed

\_\_\_\_\_  
(Number) (Country) (Day/Month/Year) Yes No

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial #)

(Filing Date)

(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; Thomas E. Tyson, Reg. No. 28,543; James H. Barksdale, Jr., Reg. No. 24,091; Casimer K. Salys, Reg. No. 28,900; Robert M. Carwell, Reg. No. 28,499; Douglas H. Lefevre, Reg. No. 26,193; Jeffrey S. LaBaw, Reg. No. 31,633; David A. Mims, Jr., Reg. 32,708; Volel Emile, Reg. No. 39,969; Anthony V. England, Reg. No. 35,129; Leslie A. Van Leeuwen, Reg. No. 42,196; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Marilyn S. Dawkins, Reg. No. 31,140; Mark E. McBurney, Reg. No. 33,114; Duke W. Yee, Reg. No. 34,285; Colin P. Cahoon, Reg. No. 38,836; Stephen R. Loe, Reg. No. 43,757; Stephen J. Walder, Reg. No. 41,534; Charles D. Stepps, Jr., Reg. No. 45,880; and Stephen R. Tkacs, Reg. No. P-46,430.

Send correspondence to: Duke W. Yee, Carstens, Yee & Cahoon, LLP, P.O. Box 802334, Dallas, Texas 75380 and direct all telephone calls to Duke W. Yee, (972) 367-2001

FULL NAME OF SOLE OR FIRST INVENTOR: Viktors Berstis

INVENTORS SIGNATURE: *Viktors Berstis* DATE: 8/30/2000

RESIDENCE: 5104 Cuesta Verde  
Austin, Texas 78746

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE

FULL NAME OF SECOND INVENTOR: Lorin Evan Ullmann

INVENTORS SIGNATURE: *Lorin Evan Ullmann* DATE: 8/17/00

RESIDENCE: 2803 Pearce Road  
Austin, Texas 78730

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE